Behavior of the Concept Lattice Reduction to Visualizing Data after Using Matrix Decompositions

Martin Polovinčák, Hussam M. Dahwa, and Václav Snášel

Department of Computer Science,
VŠB - Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava - Poruba, Czech Republic
vaclav.snasel@vsb.cz

Abstract. High complexity of formal concept analysis algorithms and lattice construction algorithms are main problems today. If we want to compute all concepts from huge incidence matrix, complexity plays a great role. In some cases, we do not need to compute all concepts, but only some of them. Our research focuses on behavior of the Concept Lattice Reduction after using matrix decompositions. Modified matrix has lower dimensions and acts as input for some known algorithms for lattice construction. In this paper we want to describe the difference between methods for matrix decompositions and describe their influence on the concept lattice.

1 Introduction

Exponentially growing for the amount of information in the past few years. Easiness using for these information and easiness access to these information, brew a big problem to retrieval of information. Also results contain a lot of data and it can be hard to make decision or find the relevant information in the result.

A lot of studies suggest that graphical representation and display of searched results can improve information retrieval performance [7][1][2][8]. A graphical information display can provide a broad and concise representation of the searched results from which can quickly comprehend their relevance and importance. The graphical information display can improve the low precision and low recall of the searched results. Other similar findings have been found in a recent study [7], in which a detailed survey is presented on how visualizations can enhance information retrieval by allowing searchers to browse through a graphical representation of the requested documents. It is therefore of utmost importance for an information retrieval system to equip with a good graphical user interface that organizes the information into an effective visual structure for the searchers to browse through during the information retrieval process.

Some researchers have proposed the use of lattice for graphical organization and visualization structuring in the construction of information retrieval systems [3]. Lattice is a network-like classification structure that can be generated automatically from a term-document indexing relationship. Such a network structure outperforms hierarchical classification structure since the former enables many paths to a particular node while the latter restricts each node to possess only

one parent. Hence lattice navigation provides an alternate browsing-based approach which can overcome the weakness of hierarchical classification browsing. However, the lattice navigation approach has its inherent problem - the curse of dimension, namely, the lattice representation of a document collection is too large to fit in a screen even for small databases. To visualize large structures, researchers have developed interfaces that allow multiple local and global views [10]. However, the method has the disadvantage that the searchers need to map different graphical representation. [4] extended the work to adopt a variant of the fisheye view technique to show individual nodes of the lattice on a standalone symbolic lisp machine. Despite these many research efforts, the display and comprehension of the lattice associated with a large database remains an open problem.

In this paper we show behavior difference of the concept lattice reduction after using various methods of matrix decompositions. These methods are well known in the area of Information retrieval under the name Latent Semantic Indexing (LSI) or Latent Semantic Analysis (LSA). LSI and LSA have been used for discovery of latent dependencies between terms (or documents). We would like to apply this approach in the area of formal concept analysis (FCA). The goal is to minimize input data before constructing the concept lattice. The reduced input data has lower dimension than the original data and computing the lattice leads in creating smaller graphic representation in less time.

We will use two methods for matrix decomposition, singular value decomposition (SVD) and Non-negative Matrix Factorization (NMF). We will mention basic terms of Formal Concept Analysis (FCA) and than we describe the rank-k of SVD and NMF decompositions. Small examples will illustrate our approach. We hope this is the way how to use FCA on large data collections and visualize data structure via concept lattice. We remind that usage of these methods depends on concrete application, because it makes some noise in the lattice structure to make it more simply.

2 Formal Concept Analyses, Concept Lattice

Definition 1. *[10] A formal context (G, M, I) consists of two sets G and M and relation I between G and M. The elements of G are called the objects and the elements of M are called the attributes of the context. In order to express that an object $ginG$ is in a relation I with an attribute $minM$, we write gIm or $(g, m)inI$ and read it as the object g has the attribute m. The relation I is also called the incidence relation of the context.*

Definition 2. *[10] For a set $A \subseteq G$ of object we define*

$$A^{'} = \{m \in M \mid gIm \ for\ all\ g \in A\}$$

(the set of attributes common to the objects in A). Correspondingly, for a set $B \subseteq M$ of attributes we define

$$B^{'} = \{g \in G \mid gIm \ for\ all\ m \in B\}$$

(the set of objects which have all attributes in B).

Definition 3. *A formal concept of the context (G, M, I) is a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A^{'} = B$ and $B^{'} = A$. We call A the extent and B the intent of the concept (A, B). $B(G, M, I)$ denotes the set of all concepts of context (G, M, I).*

Definition 4. *The concept lattice $B(G, M, I)$ is a complete lattice in which infimum and supremum are given by:*

$$\bigwedge_{t \in T}(A_t, B_t) = \left(\bigcap_{t \in T} A_t, \left(\bigcup_{t \in T} B_t \right)^{''} \right)$$

$$\bigvee_{t \in T}(A_t, B_t) = \left(\left(\bigcup_{t \in T} A_t \right)^{''}, \bigcap_{t \in T} B_t \right) .$$

.

3 Singular Value Decomposition (SVD)

Singular value decomposition (SVD) is well known because of its application in information retrieval as LSI. SVD is especially suitable in its variant for sparse matrices.

Theorem 1. *Let A is an $m \times n$ rank-r matrix. Be $\sigma_1 \geq \ldots \geq \sigma_r$ eigenvalues of a matrix $\sqrt{AA^T}$. Then there exist orthogonal matrices $U = (u_1, \ldots, u_r)$ and $V = (v_1, \ldots, v_r)$, whose column vectors are orthonormal, and a diagonal matrix $\Sigma = diag(\sigma_1, \ldots, \sigma_r)$, such that $A = U\Sigma V^{\mathrm{T}}$. The decomposition $A = U\Sigma V^T$ is called* singular value decomposition *of matrix A and numbers $\sigma_1, \ldots, \sigma_r$ are* singular values *of the matrix A. Columns of U (or V) are called* left (or right) singular vectors *of matrix A.*

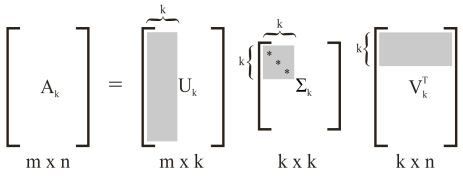

Fig. 1. k-rank SVD

Now we have a decomposition of original matrix A. It is not needed to say, that the left and right singular vectors are not sparse. We have at most r nonzero singular umbers, where rank-r is the smaller of the two matrix dimensions. Because the singular values usually fall quickly, we can take only k greatest singular values and corresponding singular vector coordinates and create a k-reduced singular decomposition of matrix A.

Definition 5. *Let us have* $k, 0 < k < r$ *and singular value decomposition of* A

$$A = U\Sigma V^T = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

We call $A_k = U_k \Sigma_k V_k^T$ *a* k-reduced singular value decomposition (rank-k SVD).

Table 1. Simplified formal context after SVD

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
O0	0	0	0	0	0	0	0	0	0	1
O1	1	0	0	0	0	0	1	0	0	0
O2	1	0	0	0	0	0	0	0	0	1
O3	1	0	0	0	0	0	0	0	0	1
O4	1	0	0	0	0	0	0	1	0	0
O5	0	0	0	0	0	0	0	1	0	0
O6	0	0	1	1	0	0	1	1	0	0
O7	1	0	0	0	0	0	0	1	0	0
O8	0	0	1	1	0	0	1	0	0	0
O9	0	0	1	1	0	0	1	1	0	0
O10	0	0	1	1	0	0	1	1	0	0
O11	0	0	1	1	0	0	1	0	0	0
O12	0	0	1	1	0	0	1	0	0	1
O13	1	0	1	1	0	0	0	0	0	1
O14	0	0	0	0	0	0	1	0	0	0
O15	1	0	1	1	0	0	1	0	0	0
O16	0	0	0	0	0	0	1	0	0	0
O17	0	0	0	0	0	0	0	1	0	0
O18	0	0	1	1	0	0	1	0	0	0s
O19	0	0	1	1	0	0	1	1	0	0

Take a closer look on $k - rank$ SVD with information retrieval motivation. If every document is relevant to only one topic, we obtain a latent semantics semantically related words (and documents) will have similar vectors in the reduced space. For an illustration of $k - rankSVD$ see figure 1. Grey areas determine first k coordinates from singular vectors, which are being used.

Theorem 2. *[11] Among all* $m \times n$ *matrices* C *of rank at most* k A_k *is the one, that minimises* $||A_k - A||_F^2 = \sum_{i,j}(A_{i,j} - C_{w,j})^2$.

Because $k - rank$ SVD is the best $k - rank$ approximation of original matrix A, any other decomposition will increase the sum of squares of matrix AA_k.

The SVD is hard to compute and once computed, it reflects only the decomposition of original matrix. The recalculation of the SVD is expensive, so it is impossible to recalculate the SVD every time new rows or columns are inserted. SVD-Updating is a partial solution, but since the error slightly increases with

Table 2. Formal concepts after SVD reduction

Node	Objects	Attributes
0	2, 3, 12, 13, 15	9
1	1, 4, 5, 6, 7, 9, 10, 17, 19	7
2	6, 8, 9, 10, 11, 12, 14, 16, 18, 19	6
3	6, 8, 9, 10, 11, 12, 13, 15, 18, 19	3, 6
4	12, 13, 15	2, 3, 9
5	6, 8, 9, 10, 11, 12, 18, 19	2, 3, 6
6	12	2, 3, 6, 9
7	6, 9, 10, 19	2, 3, 6, 7
8	1, 2, 3, 4, 7, 13, 15	0
9	2, 3, 13, 15	0, 9
10	1, 4, 7	0, 7
11	13, 15	0, 2, 3, 9
12		0 - 9
13	0 - 19	

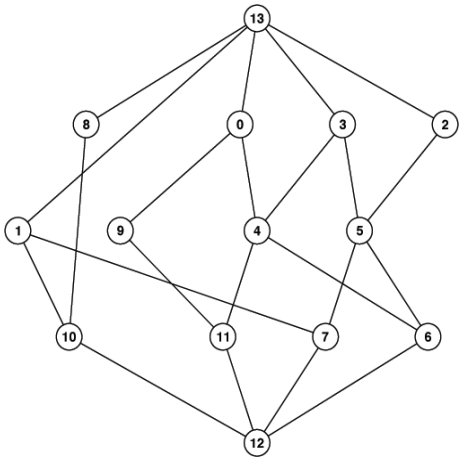

Fig. 2. Concept lattice computed from simplified formal context (Table 1)

inserted rows and columns, if the updates happen frequently, the recalculation of the SVD may be needed soon or later.

We can see that the node 3 from the original concept lattice was deleted, because the attributes composition (A6 and A9) in the objects (O1, O2) is not available because after using SVD the attribute A6 was deleted from the objects (O1,O2). And the node 20 was deleted too, because the attributes composition (A0 and A6) in the objects (O1, O4, O17) is not available because after using SVD, the attribute A6 was deleted from the original objects (O1, O4, O17).

We can see also, that after using SVD, some attributes are removed and added, and more objects have the same compositions of attributes. The node 11 has an composition of attributes (A2 and A6) in the objects (O6,O9,O12,O13), these composition of attributes (A2, A6) existed in the objects (O8, O10, O11, O18, O19), too. The node 6 has composition of attributes (A3 and A6) in the objects (O6, O10, O11, O16), these composition of attributes (A3,A6) existed in the objects (O8,O9,O11,O18,O19) too. From that, the nodes (11 and 6) are incorporated in new node (5), because all the attributes in the two nodes are existed in the all of the objects in the two nodes. That means, that the new node 5 has composition of attributes (A2, A3, A6) in the objects (O6, O8, O9, O10, O11, O12, O18, O19) (Figure 3)

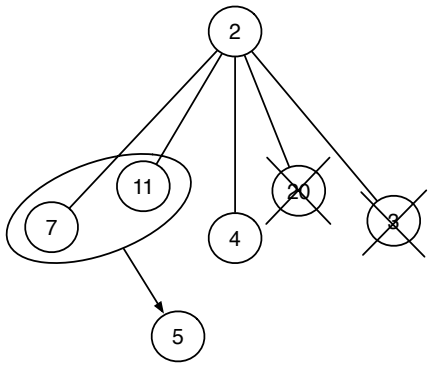

Fig. 3. Moving the nodes

4 Nonnegative Matrix Factorization (NMF)

Nonnegative matrix factorization differs from other rank reduction methods for vector space models in text mining by using of constraints that produce nonnegative basis vectors, which make possible the concept of a parts-based representation. [6] first introduced the notion of parts-based representations for problems in image analysis or text mining that occupy nonnegative subspaces in a vector-space model. Basis vectors contain no negative entries. This allows

only additive combinations of the vectors to reproduce the original. The perception of the whole becomes a combination of its parts represented by these basis vectors. In text mining, the vectors represent or identify semantic features, i.e., a set of words denoting a particular concept or topic. If a document is viewed as a combination of basis vectors, then it can be categorized as belonging to the topic represented by its principal vector. Thus, NMF can be used to organize text collections into partitioned structures or clusters directly derived from the nonnegative factors.

Table 3. Simplified formal context after NMF

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
O0	1	0	0	0	0	0	0	0	0	1
O1	1	0	0	0	0	0	1	1	0	1
O2	1	0	0	0	0	0	1	1	0	1
O3	0	0	0	0	0	0	1	1	0	0
O4	1	0	0	0	0	0	1	1	0	1
O5	1	0	0	0	0	0	0	0	0	1
O6	0	0	1	1	0	0	1	1	0	0
O7	0	0	1	0	0	0	1	1	0	0
O8	0	0	1	1	0	0	0	1	0	0
O9	0	0	1	0	0	0	1	1	0	0
O10	0	0	1	1	0	0	1	1	0	0
O11	0	0	1	1	0	0	0	1	0	0
O12	0	0	1	1	0	0	1	1	0	0
O13	0	0	1	0	0	0	1	1	0	0
O14	0	0	1	0	0	0	0	1	0	0
O15	0	0	0	1	0	0	1	1	0	0
O16	0	0	1	1	0	0	1	1	0	0
O17	1	0	1	0	0	0	1	1	0	1
O18	1	0	1	1	0	0	0	1	0	1
O19	1	0	1	1	0	0	0	1	0	1

Common approaches to NMF obtain an approximation of V by computing a (W, H) pair to minimize the Frobenius norm of the difference $V - WH$. Let $V \in R^{m \times n}$ be a nonnegative matrix and $W \in R^{m \times k}$ and $H \in R^{k \times n}$ for $0 < k << min(m, n)$. Then, the objective function or minimization problem can be stated as: $min_{W,H}||V - WH||_F^2$ with $W_{ij} > 0$ and $H_{ij} > 0$ for each i and j.

The matrices W and H are not unique. Usually H is initialized to zero and W to a randomly generated matrix where each $W_{ij} > 0$ and these initial estimates are improved or updated with alternating iterations of the algorithm. In the following subsections some existing NMF techniques are discussed.

4.1 Multiplicative method.

The NMF method proposed by Lee and Seung is based on multiplicative update rules of W and H. This scheme is referred to as the multiplicative method (MM).

MM Algorithm:

1. Initialize W and H with nonnegative values.
2. Iterate for each c, j, and i until convergence or after l iterations:
   (a) $H_{cj} \leftarrow H_{cj} \frac{(W^T V)_{cj}}{(W_T WH)_{cj}+\varepsilon}$
   (b) $W_{ic} \leftarrow W_{ic} \frac{(WH^T)_{ic}}{(WHH_T)_{ic}+\varepsilon}$

In steps 2(a) and 2(b), ε, a small positive parameter equal to 10^{-9}, is added to avoid division by zero. As observed from the MM Algorithm, W and H remain nonnegative during the updates. Simultaneous updating of W and H generally yield better results than updating each matrix factor fully before the other. In the algorithm, the columns of W or the basis vectors are normalized at each iteration; in case of W, the optimization is performed on a unit hyper sphere with the columns of W effectively being mapped to the surface of the hypersphere by repeated normalization [9]. The computational complexity of MM can be shown to be $O(kmn)$ operations (for a $rank - k$ approximation) per iteration [9].

4.2 Sparse encoding.

A new nonnegative sparse encoding scheme, based on the study of neural networks has been suggested by [5]. The method proposed by [5] has an important feature that enforces a statistical sparsity of the H matrix. As the sparsity of H increases, the basis vectors become more localized, i.e., the parts-based representation of the data in W becomes more and more enhanced.

We can see from the simplified formal context after using NMF, that we obtain new attributes that were added to the objects. This adding gives us new compositions of attributes that appear more accuracy. with showing Concept lattice from simplified formal context after using NMF, we can see, that the node 9 that has attribute a2 that existed in the objects (O6, O8, O9, O11, O12, O13, O14, O18, O19) from the original Concept lattice from simplified formal context, was got attribute a7 after using NMF, because this attribute was added to all of the objects that brew this node. Also, can see all nodes whose branched from node 9 got attribute a7 (nodes 5, 6 and 11). (Fig. 5 and 6).

In the other side, the node 1 who's has attribute a7, a new node has added to the nodes that have branched from it. This new node has a new composition of attributes (a0, a7 and a9) that existed in the objects (O1, O2, O4, O17, O18, O19) because some attributes was added to this objects after using NMF.

5 Conclusion and future work

In this paper we discussed the complexity problem in lattice-based information retrieval. We describe reduction from the mathematical point of view. The reduction process requires the setting up of an equivalence relation. We describe using two methods for obtaining the relation and we are describing decompositions result more precisely. Singular value decomposition is used as the first, non-negative matrix factorization is used as the second practical approach.

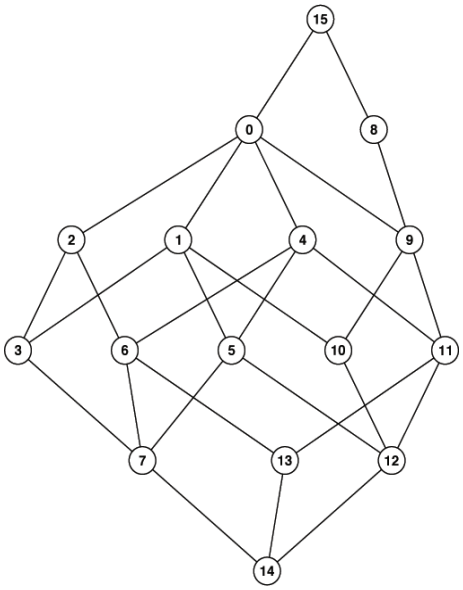

Fig. 4. Concept lattice computed from simplified formal context (Table 3)

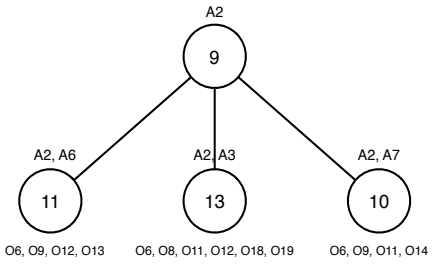

Fig. 5. Part of original Concept lattice

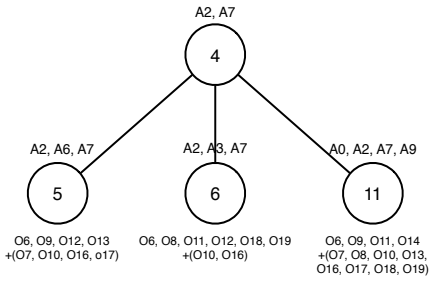

Fig. 6. Part of concept lattice after using NMF

Table 4. Formal concepts after NMF reduction

Node	Objects	Attributes
0	0, 1, 2, 18, 19	9
1	6, 7, 9, 10, 11, 14, 16, 17	7
2	1, 2, 3, 4, 6, 7, 9, 10, 12, 13, 15, 17	6
3	1, 2	6, 9
4	6, 7, 9, 10, 17	6, 7
5	6, 8, 10, 11, 12, 15, 16, 18, 19	3
6	6, 10, 11, 16	3, 7
7	6, 10, 12, 15	3, 6
8	6, 10	3, 6, 7
9	6, 8, 9, 11, 12, 13, 14, 18, 19	2
10	6, 9, 11, 14	2, 7
11	6, 9, 12, 13	2, 6
12	6, 9	2, 6, 7
13	6, 8, 11, 12, 18, 19	2, 3
14	18, 19	2, 3, 9
15	6, 11	2, 3, 7
16	6, 12	2, 3, 6
17	6	2, 3, 6, 7
18	1, 4, 5, 17, 19	0
19	1, 19	0, 9
20	1, 4, 17	0, 6
21	1	0, 6, 9
22	17	0, 6, 7
23	19	0, 2, 3, 9
24		0 - 9
25	0 - 19	

We can see that using both methods was successfull and reduced original concept lattice. The number of concepts in reduced concept lattices is lower than in case of original concept lattice. It implies that computation time of reduced lattice will be lower and that is why reducing lattices can be useful.

References

1. Bruza P.D., and Dennis S.: Query reformulation on the internet: empirical data and the hyper index search engine. Proceedings of the RIAO97 ConferenceComputer-Assisted Information Searching on Internet. (1997)
2. Bruza P.D., and McArthur R.: Interactive internet search: Keyword, directory and query reformulation mechanisms compared. Proceedings of the 23rd Annual ACM Conference of Research and Development in Information Retrieval (SIGIR 2000), ACM Press, (2000)
3. Carpineto C., Romano G. ULYSSES: A lattice-based multiple interaction strategy retrieval interface. Gornostaev J, Blumenthal B and Unger C, Eds., Lecture Notes in Computer ScienceHuman-Computer Interaction proceedings of a conference in Berlin in 1995, Springer-Verlag, Berlin, (1995) 91-104
4. Carpineto C., and Romano G.: A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning, (1996) 24:95122.
5. Hoyer P.: Non-negative Sparse Coding. Proceedings of the IEEE workshop on neural networks for signal processing. Martigny, Switzerland, (2002)
6. Lee D., Seung H.: Algorithms for non-negative matrix factorization. Advances in eural Information Processing Systems, (2001) 556–562.
7. Lin X.: Map displays for information retrieval. Journal of the American Society of Information Science, (1997) 48:40-54
8. McArthur R., Bruza P.D.: The ranking of query refinements in interactive web-based retrieval. Proceedings of the Information Doors Workshop (Held in conjunction with the ACM Hypertext and Digital Libraries Conferences), (2000)
9. Pauca V., Shahnaz F., Berry M., Plemmons, R.: Text mining using nonnegative matrix factorizations. In Proceedings of the fourth SIAM international conference on data mining, April 22 24. SIAM, Lake Buena Vista, FL, (2004)
10. Wille R.: Lattices and data analysis: How to draw them using a computer. Rival I, Eds., Algorithms and Order. Kluwer, Boston, (1989) 33-58.
11. Young G., Eckart C.: The approximation of one matrix by another of lower rank. sychometrika 1, (1936) 211–218.